
Personal & Business Systems Division

Advanced SSL support in Comet

Group: Comet Team

Specification for advanced SSL support in Comet

Owner – Ron Mondri (RonMond)

Version 0.4

Last Saved: [REDACTED]

Status: DRAFT

Distribution: Microsoft Internal Distribution

© Copyright Microsoft Corporation, 2000. All Rights Reserved

Microsoft Confidential

Note: This documentation is an early release of the final product documentation. It is meant to accompany software that is still in development. Some of the information in this documentation may be inaccurate or may not be an accurate representation of the functionality of the final retail product. Microsoft assumes no responsibility for any damages that might occur either directly or indirectly from these inaccuracies.

1.	OVERVIEW	3
2.	REQUIREMENTS	4
2.1.	GENERAL	4
2.1.1.	<i>Required features</i>	4
2.1.2.	<i>About web server certificates</i>	4
2.1.3.	<i>Hosting multiple sites behind Comet as reverse proxy</i>	4
2.1.4.	<i>Comet array considerations</i>	5
2.1.5.	<i>Coexistence with header based authentication</i>	5
2.2.	SSL SESSION TERMINATION	6
2.2.1.	<i>Description</i>	6
2.2.2.	<i>Installing the server certificates</i>	6
2.3.	SSL SESSION BRIDGING (DUAL-HOP SSL)	6
2.3.1.	<i>Description</i>	6
2.3.2.	<i>Client SSL authentication</i>	7
2.3.3.	<i>Support for bridging functionality</i>	7
2.4.	CONSIDERATIONS FOR FORWARD PROXY	7
3.	USER INTERFACE	8
3.1.	CERTIFICATE MANAGEMENT	8
3.1.1.	<i>Configuring server certificates</i>	8
3.1.2.	<i>Configuring Comet client certificate</i>	8
3.2.	SSL IN COMET RULES	8
3.2.1.	<i>Configuring SSL in web publishing rules</i>	8
3.2.2.	<i>Configuring SSL in routing rules</i>	11
3.2.3.	<i>Choosing certificates</i>	12
3.3.	CONFIGURING PROXY IP ADDRESS PROPERTIES	12
3.3.1.	<i>Array properties dialog</i>	12
4.	REVISION HISTORY	18

1. Overview

NOTICE

This document is an extremely early DRAFT. It has not yet been reviewed or approved, nor have priorities been properly assigned in all circumstances. Please keep this in mind when reading this document.

Additional feature suggestions and comments can be emailed to comet@microsoft.com.

SSL is the primary protocol for secure transactions in the Internet. It involves the establishment of an encrypted session between the client and server, and supports identification mechanisms that optionally let the two sides authenticate each other.

Conventional HTTP proxies handle SSL sessions by terminating the SSL sessions at the proxy and forwarding clear HTTP requests to the web server. Recently, there has been a growing demand for advanced SSL support in the proxy, including various requirements on top of terminating the SSL session at the proxy, such as supporting dual-hop SSL sessions, and SSL authentication between proxy and server.

Microsoft Proxy Server 2.0 provided session termination at proxy, as a part of (integrated) IIS. In Comet, it is required both to provide this functionality as a part of Comet due to the separation from IIS, and to support the more advanced features as will be described herein.

This specification defines the requirements for advanced SSL features in Comet, and the associated user interface.

2. Requirements

2.1. General

2.1.1. Required features

Comet will provide the following advanced SSL features:

- SSL session termination at Comet server
- SSL session bridging (dual-hop SSL)
- Authentication (validation) of Comet server using certificate
- Authentication (validation) of SSL server, in Comet, using certificates

The functionality will be mirrored for reverse proxy and for forward proxy. While reverse proxy scenarios that require this kind of treatment are common, using it in forward proxy is a new concept. Still, it is a customer requirement.

2.1.2. About web server certificates

Web server certificates are issued by (trusted) certificate authorities and stored on the web site for the SSL session negotiation. The web client (such as IE) initiates the SSL session negotiation, during which the server identifies itself by presenting the stored certificate to the client.

When a certificate is issued to an organization the DNS name of the web site is a part of the signature (the common name field). This is how the web client authenticates the web site certificate - it compares the name that was requested by the user to the name in the certificate. For a more comprehensive description of this process, refer to the chapter "Validating Certificates" in MSDN.

This method enables serving the site from multiple IP addresses, as long as these share the same DNS name.

2.1.3. Hosting multiple sites behind Comet as reverse proxy

Comet allows multiple sites to be hosted behind the reverse proxy. This is accomplished by defining multiple web publishing rules, each one corresponds to a hosted site.

When SSL session termination at Comet is required, the SSL server that the client sees is in fact the Comet server. Comet will therefore have to identify as the web server by using its certificate. When Comet hosts several web sites, it is unfortunately impossible to know which server the request should be routed to at the phase where the SSL session is established. The only way to solve this problem is by adding multiple IP addresses to the Comet server external interface(s), where each address corresponds to a hosted web site. A certificate will then be mapped per IP address, and this certificate will be used for the SSL server authentication.

This implies a limitation on current web publishing design, since the certificates cannot be a part of the web publishing rule in its current form. Due to the reasons mentioned above, the certificates will be configured in the UI as a property of the server related to addresses (see below for full description of implementation).

2.1.4. Comet array considerations

Comet enables serving the same web site from multiple machines through the array functionality. For SSL and certificate purposes, it is irrelevant since the server authentication scheme is not dependant on IP addresses. However, if the administrator does not add the relevant certificate to a server in the array, SSL calls to that server will fail.

Comet admin could have potentially validated a configured certificate's existence by querying all servers in the array and not allowing setting a certificate that is not present in all of the servers. This would have been similar to Dialup entry validation when a demand dial phonebook entry is configured from the admin. We will not implement this feature in this release due to lack of resources. It is therefore up to the user to make sure that the configured certificate exists in all servers in the array.

2.1.5. Coexistence with header based authentication

In IIS, the user may select a number of header based authentication schemes to apply to the web site: Basic, Digest, and Integrated Windows Authentication. Comet provides the same functionality for web sites hosted behind the reverse proxy. Two issues are of specific importance here: integration between the multiple authentication schemes, and integration with Comet rule structure.

When a web requests enters the reverse proxy, it is checked against the existing rule base as anonymous. If there is an existing rule for the destination of the request that allows anonymous access, then the request is processed. Otherwise, there is need to authenticate the client. Since both certificate based authentication and header authentication may be applied, defining authentication method on a per-rule basis may lead to conflicting rules that will force SSL renegotiation for client certificate as well as header authentication. This behavior causes redundant round trips for the common case (header authentication only) and is generally not well defined in web clients.

It is safe to assume that in the common case, authentication method will be single for a given web site. Since Comet may host multiple sites, we have to provide the functionality and the flexibility needed to support hosting scenarios, but not to neglect the simplicity needed to support the common case of a single hosted site.

The "natural" place to set authentication configuration in is the publishing rule. Still, an important flexibility that we do not want to lose is the ability to direct requests to alternate sites based on who the authenticated user is. This would lead to the aforementioned conflicts between rules; hence authentication method must be defined and applied on a higher level, before the request enters the rule evaluation phase.

In Comet beta, header authentication was defined for the entire reverse proxy (all listeners). This limited the flexibility in hosting environments. A more flexible approach would be to assign these properties to listeners (defined by reverse proxy IP address port pairs). It is still important to support the common and simple case where the reverse proxy listens on all IP addresses. This is supported today in the scheme by assigning an "all IP addresses" entry for the reverse proxy. This functionality has to be preserved.

The integrated approach that will be presented here includes the definition of authentication properties for proxy listeners. This includes header based as well as certificate based authentication.

2.2. SSL session termination

2.2.1. Description

Comet will allow users to configure their server such that SSL requests coming from clients will be terminated at the Comet server. Naturally, in "normal" Internet context this scenario is only applicable for corporate reverse proxies. This is due to the fact that when Internet browsers establish SSL sessions they use the server certificate to validate that the server that replied is indeed the target server. If the session is terminated in some intermediate proxy, the certificate will not match and the user will be notified that the server is not who it claims to be.

In reverse proxy scenarios, where the web server owner can configure the proxy to use his own certificate, terminating SSL at the proxy is a possibility.

There are two reasons for performing such a task at the proxy level:

1. Offloading SSL computation from the web server to the reverse proxy (subsequently reducing CPU load from the web server)
2. Not allowing tunnels through the organization, except those created by the proxy/firewall.

In Proxy Server 2.0, this functionality was provided by IIS as a part of its site definition interface. In Comet, we have to provide that functionality as well. As explained in the previous section, we will not use publishing rules as the configuration point for the certificates, although that option is the natural choice.

2.2.2. Installing the server certificates

Comet will not provide the functionality of actually installing the server certificates on the Comet machines in the array (lack of development time). It will be the administrator's job to obtain these certificates and to add them to the local certificate store.

Comet will provide, however, means of enumerating these certificates that exist on the machine as an aid to the administrator when coming to configure which certificate will be mapped to which external IP address.

As previously discussed, the administrator has to be aware that bad configuration scenarios may occur if a certain server certificate is not installed on one or more machines in the array. Presence of matching certificates will not be validated in Comet. One may consider this a feature of allowing asymmetric hosting situations where different subsets of the array will host different web sites. However, if the administrator then adds all array external IP addresses to the DNS entry of the site's name, some of the requests will fail.

2.3. SSL session bridging (dual-hop SSL)

2.3.1. Description

SSL session bridging means that the Comet server will terminate the SSL connection and establish another SSL session, as a client, to the web server. This essentially means that an SSL client will be implemented in Comet.

This requirement is an enhanced version of session termination. It will be desired whenever termination is desired, and encrypted communications in the corporate (between reverse proxy and web server) is also required. In this scenario Comet acts as both SSL server and SSL client.

2.3.2. Client SSL authentication

It is possible to establish an SSL session without using a client certificate. This is the common scenario in Internet browsing. For the RP case, however, it is sometimes important to maintain connection integrity by providing extra authentication between the reverse proxy and the web server. Comet will therefore provide the ability to enable such client authentication in second hop SSL sessions, and the ability to assign a certificate to this activity.

As in the server certificate case, Comet will have a limited functionality in handling the certificates. The administrator will have to add the certificate to the machine, and Comet admin will only have the ability to assign an existing certificate for the Comet client to use.

2.3.3. Support for bridging functionality

For supporting a complete session bridging solution, the following items have to be implemented in Comet:

1. Full SSL client functionality (handling SSL protocol client side, authentication of certificates using pre-configured trusted CA list, etc.)
2. Controls that let the administrator choose how the second hop should be processed. All four possible "switching" options are required:
 - o HTTP -> HTTP [normal operation]
 - o SSL -> HTTP [session termination]
 - o SSL -> SSL [session bridging - dual hop SSL]
 - o HTTP -> SSL [encryption for remote reverse proxy]
3. Controls that allow enabling of client authentication in the second hop SSL connections.
4. Certificate maintenance – assigning a client certificate to the Comet server for using in case client authentication is required

2.4. Considerations for forward proxy

As explained before, Comet will provide the same functionality for forward proxy. This will both answer customer requirements, and be in line with the engine mirrored functionality for reverse and forward proxy.

All details of the discussion above also apply to forward proxy. The same functionality has to be mirrored in this case. Naturally, there is no "hosted site" entity in forward proxy. Instead, we can treat the uplink route as the site in the publishing case. The destination filter is the same: a request arrives and is matched against a destination set. If there is a match, it is routed upstream. All the advanced SSL functionality should be applied to this upstream link.

3. User Interface

3.1. Certificate management

3.1.1. Configuring server certificates

Server certificates will be configured per listener, on per group of listeners in case "All IP addresses" is selected for the assignment of certificates.

The UI for doing this mapping will be exposed in the new design of the reverse proxy and forward proxy listener properties dialogs. See the section "Proxy properties".

3.1.2. Configuring Comet client certificate

For reverse proxy, a client certificate may be configured per publishing rule, in order to allow using a different certificate as identification to different web servers. Note that this flexibility is not expected to be used in the common case, but it is possible that demand may rise.

As in the server certificate case, there will be no method provided to actually install a certificate on the server or to configure properties such as supplied by IE. The administrator will have to import certificates using the server certificate management snap-in. Comet admin UI will let the administrator select a certificate from the list of available certificates on the server.

In the forward proxy case, a mirrored functionality will be provided, and definition of a client certificate may be done per routing rule. Note that in this case, this means that the proxy will identify to the next hop server (proxy or web server) using this certificate.

For the specific UI definition, see the following section, which defines web publishing and routing rules and its "choosing certificates" subsection.

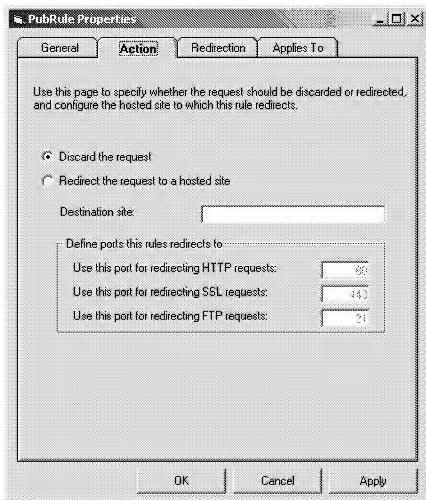
3.2. SSL in Comet rules

3.2.1. Configuring SSL in web publishing rules

The SSL configuration UI that enables session termination and session bridging will be integrated into publishing rule properties. This will allow setting these properties per hosted site, which is the most flexible behavior. The only part of SSL handling that will not be controlled from the rule properties is the server certificates that will have to be configured as a server property, related to the IP address. See the discussion about that in the previous section.

The administrator will be able to modify SSL settings from a new tab of the publishing rule, called "Redirection". The "Action" tab will also change a bit, and will now expose only basic functionality. The reasoning behind this design is that SSL bridging is an advanced feature, yet useful for some important customers. We do not want to further complicate the configuration for the common case, but we also don't want to hide this feature and we want it to get good exposure. The current design places only essential configuration parameters in the "Action" tab, and shows clear defaults for the "Redirection" tab.

The new "Action" tab will look like this:

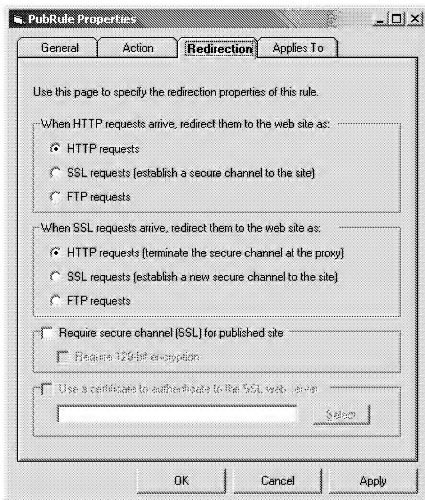


Default values for controls in this tab:

#	Control	Value
1	Top level radio button (Discard / Redirect)	As user chose when creating rule
2	Destination site edit box	Empty
3	HTTP port edit	80
4	SSL port edit	443
5	FTP port edit	21

The "Destination site" edit box and the port edit boxes are disabled if the radio calls for discarding the request.

The new "Redirection" tab will be as follows:



The “when HTTP requests arrive” radio button group on this dialog allows an advanced feature of encrypting incoming HTTP traffic between the proxy and the web server, or redirection to an FTP server instead of continuing with HTTP. While second hop encryption seems a little bit odd at first, we want to support this feature for the scenario where the proxy is close to the client and not to the web server and we would like encrypting the second hop (and the code supports that anyway).

The “when SSL requests arrive” radio button group defines whether Comet will simply do a session termination (the default, same as in Proxy 2.0), create a second hop SSL, or terminate the SSL session and continue to an FTP site.

Note that in both of these groups, the terms used are not exactly technically correct, since all are HTTP requests, and the distinction should be between a secure channel to an insecure one. However, for simplicity of the user interface the proposed wording seems to be better.

The “require SSL” checkbox, when checked, will define that the published site will only accept secure channel requests (only incoming requests on a secure channel will be handled). If the “128 bit” box is also checked (enabled when the first box is checked) then only 128-bit encrypted sessions will be accepted.

In the third group, the user may select a client certificate to use for the purpose of authenticating the proxy to the web server. The available client certificates will be enumerated and shown in a separate dialog when the user presses the “Select” button. After the user selects a certificate, the

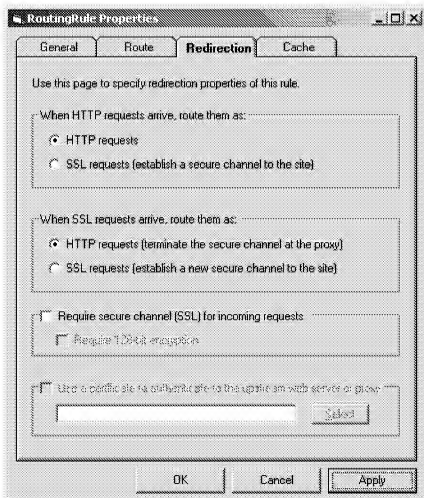
edit box will show the certificate's "friendly name". The "choosing certificates" section describes the certificate enumeration dialog.

Default values for controls on this tab are as seen in the mockup. Additionally, if the "Require SSL" checkbox is checked, the first group of radio buttons (HTTP redirection) should be disabled since it has no meaning in that case. The "Use a certificate" checkbox will only be enabled if the user has selected one of the "SSL requests" buttons in an enabled tri-button redirection group.

3.2.2. Configuring SSL in routing rules

The functionality provided in routing rules would be an exact mirror of the one offered in web publishing rules. The forward proxy will filter requests based on destination, and while routing them forward it will apply the advanced SSL settings to the requests.

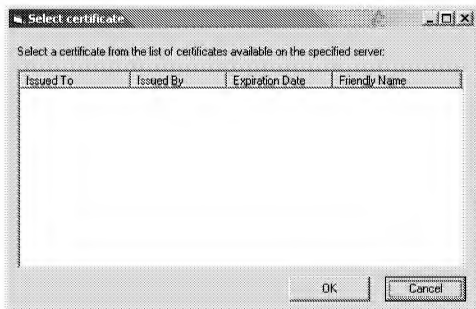
Routing rules will have an additional "Redirection" tab similar to the publishing rule tab of the same name, except that requests can only be sent out as HTTP or SSL, and not as FTP. Also, the text here is a little bit different to accommodate to the different usage scenario:



The same explanations, default values, and control enabling behavior given for web publishing rules apply in this case.

3.2.3. Choosing certificates

The following dialog box will be used to configure certificates. This is a simplified version of the standard certificate selection dialog box. It enables the user to select a certificate from the given list (which is an enumeration of existing certificates on the machine), but has no functionality relating to certificate import or configuration.



On startup of the dialog, the local certificate store will be enumerated and all certificates presented in the list box. The user must highlight a single row in the list box before the OK button is pressed, in order to make the selection.

Note that this dialog is generic and should be applied whenever a certificate should be chosen for subsequent usage in Comet. Whatever the need: server certificates, client certificates, Comet admin will select a different store to retrieve the certificates from, but the same dialog will be used.

3.3. Configuring proxy IP address properties

3.3.1. Array properties dialog

As described in the requirements section, authentication configuration will be made from the listener level and not from the proxy level.

Additional functionality defined below:

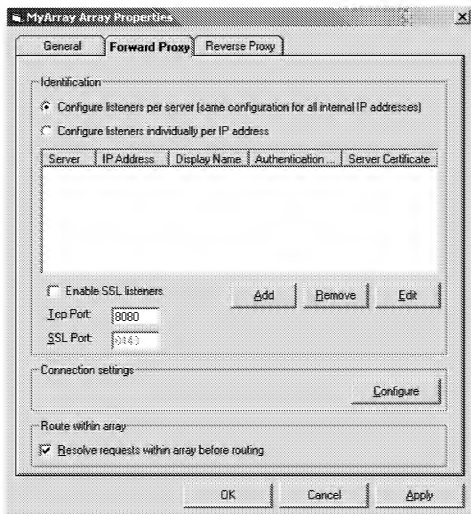
- Define authentication method per listener IP
- Add client certificate as a valid authentication method
- Add a display name property and a server certificate property per IP listener (see explanation below)
- Add the ability to configure server certificate per reverse proxy IP address
- Add the same functionality for forward proxy

If Comet hosts a single site, or if no SSL is required (no server certificate needed) there are no special requirements for the configuration of the reverse proxy. In this case, the reverse proxy should be configured to use all IP addresses as listeners, and no distinction will be made

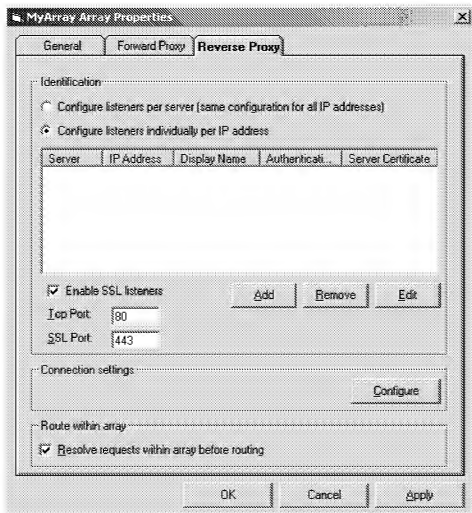
between them: a server certificate may be added to the "all IP" entry, and an associated display name will be given.

If Comet is used in a hosting environment and publishes multiple sites, and the sites require SSL functionality, then the sites must be allocated individual IP addresses. This is due to the SSL session establishment protocol described in a previous section. To accommodate to such flexibility and provide a useful UI, Comet administrator will be able to add display names to "IP listeners" (effectively IP addresses). If a site is published on more than a single IP (for example, in the case of an array), the administrator is expected to use the same display name for the IP. This will allow sorting of the listener list based on display name and applying the same configuration changes to all the relevant IP addresses.

The array properties dialog is available through selecting properties on the array node. The "Forward proxy" tab will be as follows:



The "Reverse proxy" tab will be very similar. Naturally, the displayed properties are distinct to the reverse proxy case:



For both tabs, default values are shown.

The topmost radio buttons ("identification" group) determine whether the listener properties should be set for all IP addresses (all internal IP addresses in forward proxy) or for individual IP addresses. This property is array wide, which means that if an administrator will not be able to configure some servers using "all IP addresses" and some "individually". This is not an operational limitation, since this functionality can be reached by selecting the "individual" setting and setting the same properties for all IP addresses of the machines for which it is desired.

The configuration of the individual IP address properties will be kept valid even when the user switches to "all IP" mode, and vice versa. This means that a "shadow configuration" will be kept on this specific item. The reason is that it is a fairly complex setting to make and we do not want it to be lost because of a simple administrator's configuration error. Of course, entries can be removed using the "remove" button.

Default entries: For forward proxy, the default entry in the listbox would be "All internal IP addresses" with the default configuration options (see below for those). For reverse proxy, the default would be no entries, which means no active listeners.

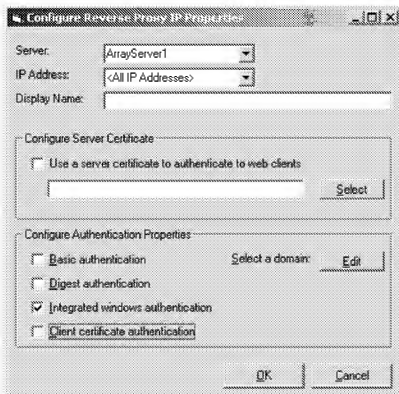
Note that for each proxy (reverse and forward), the listener listbox displays all the required information:

- Server name
- IP address

- Display name
- Authentication method
- Server certificate

In both reverse and forward proxy tabs, the "Add" button will always be enabled. "Edit" and "Remove" will only be enabled when the administrator selects an entry in the existing list.

For reverse proxy, if the administrator clicks the "Add" button, the following dialog will be displayed (default values shown in mockup):



In this dialog, the administrator must first select a server. Once a server is selected, and depending on the operation mode ("all IPs" or "individual IPs") the IP address combo box will show the available selection for listener IP addresses. In the "All IP addresses" mode, the IP address listbox will be disabled and will display "All IP Addresses". In the "individual IP addresses" mode, all IP addresses of the server (enumerated locally and retrieved by RPC) will be available in the IP address combo box..

When the dialog is opened using the "Edit" button, the "Server" and "IP address" combos will be read only. When opened using "Add", the combos will be modifiable (still, the IP address combo will be disabled for the "All IP" mode).

For the pair server+IP the administrator can specify the display name, whether to use a server certificate for authentication and which server certificate to use, and which authentication method to use for incoming requests.

When a user selects a server+IP pair, then if the administrator already defined this pair's properties, the remaining controls will be filled with that information. The user will be able to change it if needed.

Note that all four options are Boolean flags for the authentication method. This implies that an administrator may choose to enable authentication both through certificates and headers. In this

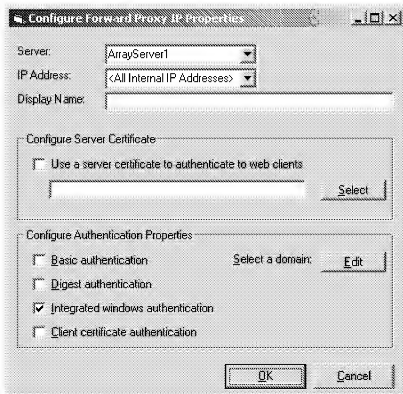
case, Comet will always try to renegotiate SSL to get the client certificate. If no such client certificate exists on the client, Comet will attempt header authentication.

In IIS, the client certificate functionality is defined through defining one of the following options: Ignore, Accept, and Require. In Comet's case, these options are mapped as follows:

- Ignore -> "Client certificate" not checked
- Accept -> "Client certificate" is checked, as well as one or more header based methods
- Require -> Only "Client certificate" is checked

The forward proxy IP configuration dialog is very similar to that of the reverse proxy. The only difference is that for forward proxy the "published" IP addresses are the internal addresses. Due to that fact, the following changes should take place:

- The entry that will be displayed in the "All IP addresses" mode is "All Internal IP addresses" instead of "All IP addresses" as in the reverse proxy case.
- The enumerated addresses in the "individual IP addresses" mode will be only the internal IP addresses as defined on the server.

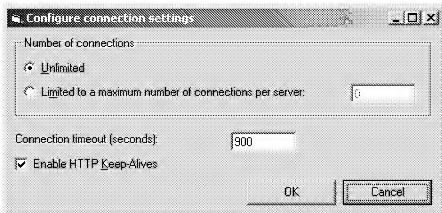


In both properties dialog, the "Select" will open the certificate selection dialog defined in the previous section. In both, the "Edit" button will open the domain selection box same as in the current implementation for authentication properties. The edit button will only be enabled if the "Basic" or "Digest" authentication schemes are selected.

3.3.2. Connection settings dialog

The connection settings functionality will be available for modification for RP and FP, separately, by pressing the appropriate button on the RP or FP dialog tab.

From the connection dialog, the administrator can configure the maximum number of connections, the connection timeout, and whether to enable HTTP keep-alive functionality. Following is a drawing of the connection setting dialog (default values displayed in mockup):



4. Revision History

30/3/2000	RonMond	V 0.4	Move all-IP listener flag to array level. Change some dialog designs. Remove FTP in fwd proxy.
20/3/2000	RonMond	V 0.3	Listener based authentication configuration, functionality in fwd proxy, change rule designs.
12/3/2000	RonMond	V 0.2	Add require SSL option to web publishing rule
7/3/2000	RonMond	V 0.1	First draft